

AMENDMENTS TO THE CLAIMS

This listing of claims replaces all prior versions, and listings, of claims in the application:

1 1. (Currently Amended) A method of distributing workload in a workflow management
2 system comprising the steps of:
3 during a calibration mode, executing plural instantiations of a test process to identify load
4 index parameters;
5 calculating a load index based on the load index parameters for each of a plurality of
6 engines ~~engine~~ of the workflow management system, wherein each load index reflects a
7 workload of its associated engine, wherein the load index corresponds to an average activity
8 execution delay; and
9 distributing workload across the plurality of engines in response to the load indices in a
10 load sensitive mode.

1 2. (Previously Presented) The method of claim 1:
2 wherein identifying the load index parameters comprises identifying a single engine
3 nominal activity execution delay (C) when no concurrent activities are executing and an activity
4 execution latency factor (λ), wherein λ is a function of a number of concurrently executing
5 activities.

1 3. (Previously Presented) The method of claim 2 wherein calculating the load index
2 comprises calculating the load index for each engine j as a total average activity execution delay

3
$$L(j) = C + \frac{1}{k} \sum_{i=1}^k N_i \lambda_i$$
, wherein k is a total number of activities completed within a

4 pre-determined time period for engine j , wherein N_i is the number of other concurrently
5 executing processes at the time activity i is executing, wherein λ_i , is an execution latency rate
6 for activity i .

- 1 4. (Previously Presented) The method of claim 2 wherein calculating the load index
2 comprises calculating the load index for each engine j as a relative average activity execution
3 delay $L(j) = \frac{1}{k} \sum_{i=1}^k N_i \lambda_i$, wherein k is a total number of activities completed within a pre-
4 determined time period for engine j , wherein N_i is the number of other concurrently executing
5 activities at the time activity i is executing, wherein λ_i is an execution latency rate for activity i .
- 1 5. (Previously Presented) The method of claim 1 wherein distributing the workload
2 comprises re-directing incoming process requests to another engine.
- 1 6. (Previously Presented) The method of claim 1 wherein distributing the workload
2 comprises re-distributing queued processes to another engine.
- 1 7. (Previously Presented) The method of claim 1 wherein distributing the workload
2 comprises prioritizing a source engine for distributing workload from based on a maximum
3 differential workload.
- 1 8. (Currently Amended) The method of claim 1 wherein distributing the workload
2 comprises identifying a target engine to which workload is to be distributed ~~for distributing~~
3 ~~workload to~~ based on a maximum differential workload.
- 1 9. (Original) A method of distributing workload in a workflow management system
2 comprising the steps of:
3 a) calculating a load index for each engine of the workflow management system,
4 wherein each load index reflects a workload of its associated engine;
5 b) operating in a load insensitive workload distribution mode for distributing
6 processes until a maximum differential load index exceeds a pre-determined threshold; and
7 c) operating in a load sensitive workload distribution mode for distributing processes
8 until all processes have completed execution once the maximum differential load index exceeds
9 the pre-determined threshold.

1 10. (Original) The method of claim 9 wherein processes are round-robin distributed in the
2 load insensitive workload distribution mode.

1 11. (Original) The method of claim 9 wherein step a) further comprises the step of
2 calculating the load index for each engine j as a total average activity execution delay
3 $L(j) = C + \frac{1}{k} \sum_{i=1}^k N_i \lambda_i$, wherein k is a total number of activities completed within a
4 pre-determined time period for engine j , wherein N_i is the number of other concurrently
5 executing processes at the time activity i is executing, wherein λ_i is an execution latency rate for
6 activity i , wherein C is a single engine nominal activity execution delay when no concurrent
7 activities are executing.

1 12. (Original) The method of claim 9 wherein step a) further comprises the step of
2 calculating the load index for each engine j as a relative average activity execution delay
3 $L(j) = \frac{1}{k} \sum_{i=1}^k N_i \lambda_i$, wherein k is a total number of activities completed within a pre-determined
4 time period for engine j , wherein N_i is the number of other concurrently executing activities at
5 the time activity i is executing, wherein λ_i is an execution latency rate for activity i .

1 13. (Original) The method of claim 9 wherein step c) further comprises the step of
2 re-directing incoming process requests to another engine.

1 14. (Original) The method of claim 9 wherein step c) further comprises the step of
2 re-distributing queued processes to another engine.

1 15. (Original) The method of claim 9 wherein step c) further comprises the step of
2 prioritizing a source engine for distributing workload from based on a maximum differential
3 workload.

1 16. (Original) The method of claim 9 wherein step c) further comprises the step of
2 identifying a target engine for distributing workload to based on a maximum differential
3 workload.

1 17. (Currently Amended) A method of distributing workload in a workflow management
2 system comprising the steps of:

3 a) switching from a load insensitive workload distribution mode to a load sensitive
4 workload distribution mode for distributing processes when a maximum differential load index
5 exceeds a first pre-determined threshold, T1; and

6 b) switching from the load sensitive workload distribution mode to the load
7 insensitive workload distribution mode for distributing processes when the maximum differential
8 load index is less than a second pre-determined threshold, T2.

1 18. (Previously Presented) The method of claim 17 wherein T1=T2.

1 19. (Previously Presented) The method of claim 17 wherein T1>T2.

1 20. (Original) The method of claim 17 wherein step a) further comprises the step of
2 calculating a load index for each engine j as a total average activity execution delay

3
$$L(j) = C + \frac{1}{k} \sum_{i=1}^k N_i \lambda_i$$
, wherein k is a total number of activities completed within a

4 pre-determined time period for engine j , wherein N_i is the number of other concurrently
5 executing processes at the time activity i is executing, wherein λ_i is an execution latency rate for
6 activity i , wherein C is a single engine activity nominal execution delay when no concurrent
7 activities are executing.

21. (Original) The method of claim 17 wherein step a) further comprises the step of calculating a load index for each engine j as a relative average activity execution delay $L(j) = \frac{1}{k} \sum_{i=1}^k N_i \lambda_i$, wherein k is a total number of activities completed within a pre-determined time period for engine j , wherein N_i is the number of other concurrently executing activities at the time activity i is executing, wherein λ_i is an execution latency rate for activity i .

22. (Previously Presented) The method of claim 1, further comprising providing a definition of activities in the test process such that for each activity, a resource execution time is much less than an engine execution time, the resource execution time representing an execution time of a resource to perform work represented by the respective activity, and the engine execution time representing an execution time of the respective engine in performing the activity.

23. (Previously Presented) A workflow management system, comprising:
plural workflow engines;
workload monitors to compute load indices for the workflow engines, wherein each load index reflects a workload of the corresponding workflow engine; and
a load balancer to:
operate in a load insensitive workload distribution mode for distributing processes among the workflow engines in a first distribution fashion that is insensitive to the load indices until at least one difference between load indices of the workflow engines exceeds a first threshold; and
after the at least one difference between load indices exceeds the first threshold, operate in a load sensitive workload distribution mode for distributing processes among the workflow engines in a second distribution fashion that is sensitive to the load indices until at least one of:
(1) all processes have completed execution; and
(2) the at least one difference between load indices of the workflow engines is less than a second threshold.

1 24. (Previously Presented) The workflow management system of claim 23, wherein the load
2 index for each engine j is a total average activity execution delay $L(j) = C + \frac{1}{k} \sum_{i=1}^k N_i \lambda_i$, wherein
3 k is a total number of activities completed within a pre-determined time period for engine j ,
4 wherein N_i is the number of other concurrently executing processes at the time activity i is
5 executing, wherein λ_i is an execution latency rate for activity i , wherein C is a single engine
6 activity nominal execution delay when no concurrent activities are executing.

1 25. (Previously Presented) The workflow management system of claim 23, wherein the load
2 index for each engine j is a relative average activity execution delay $L(j) = \frac{1}{k} \sum_{i=1}^k N_i \lambda_i$, wherein k
3 is a total number of activities completed within a pre-determined time period for engine j ,
4 wherein N_i is the number of other concurrently executing activities at the time activity i is
5 executing, wherein λ_i is an execution latency rate for activity i .

1 26. (New) The method of claim 1, wherein the plural instantiations of the test process are
2 executed during the calibration mode to increase loading on each workflow engine to enable
3 identification of the load index parameters.